数理解析研究所講究録
1166 巻 2000 年 235-243

Quantum cryptanalysis of block ciphers

Akihiro Yamamura [*] Hirokazu Ishizuka [†]

Abstract

Grover invented a quantum algorithm that finds a solution in only $O(\sqrt{2^n})$ steps whereas the exhaustive search algorithm needs $O(2^n)$ steps on average. Brassard, Høyer, Tapp construct an algorithm that counts the number of the solutions for a searching problem. We discuss two applications of quantum algorithms to information security; the first is the cryptanalysis of block ciphers using Grover's algorithm and the second is the strength evaluation of block ciphers using Brassard, Høyer, Tapp's algorithm.

1 Introduction

Quantum information processing has attracted a great attention. Quantum cryptography, quantum protocols, quantum teleportations and quantum computations are the core of quantum information processing. These technologies have a huge potentiality that they substantially exceed the existing technologies. In computer science, the Turing machine is considered a standard model of an algorithm. Numerous models for an algorithm are known to be equivalent to Turing machines in computing power. In fact, Church's thesis claims that any algorithm is realized by a Turing machine. On the other hand, an algorithm for practical computation does not correspond to a Turing machine. A plausible notion for realistic computation is the bounded error probabilistic polynomial time algorithm (**BPP**). In 1982, R.Feinmann [11] pointed out that exponential slowdown occurs when quantum physical phenomenon is simulated by a digital computer. His observation caused the conjecture that quantum physical phenomena can be employed to attain exponential speed-up in computation and many researches have been done forward the conjecture [2, 4, 12, 17].

No probabilistic polynomial time algorithm for prime factorization and the discrete logarithm problem is known up to date. The security of many public key cryptosystems are based on this fact. We do not know whether or not these problems are in the class **BPP**, however, P.Shor [16] showed that these problems are located in the class **BQP**, that is to say, these problems can be solved in polynomial time by a quantum computer. A quantum computer is implemented if we can control the quantum physical phenomena like superposition, interference and entanglement. The control of the quantum physical phenomena may be plausible, and hence, **BQP** is probably the class of realistic computation. If this is the case, many existing cryptosystems are vulnerable to attacks using quantum computers. We note that $\mathbf{BPP} \subset \mathbf{BQP} \subset \mathbf{PSPACE}$ [5]. Nothing about the relation between **NP** and **BQP** is known. Although the result by Bennett, Bernstein, Brassard, Vazirani [3] indicates $\mathbf{NP} \not\subset \mathbf{BQP}$, it does not rule out $\mathbf{NP} \subset \mathbf{BQP}$. Quantum physical phenomena may provide new computational complexity classes and contribute substantially to information processing in the future. Surprisingly, some researchers reported that even an **NP** complete problem can be solved in polynomial time by if *non-linearity* of quantum physics can be employed. This implies that no public key cryptosystem is secure if this is the case. However, the claim has not been officially accepted by researchers in the field.

In this paper, we show that for any block cipher of key size n, one can mount a known-plaintext attack that finds the secret encrypting key in $O(\sqrt{2^n})$ steps using Grover's algorithm, that is to say, the attacker has an arbitrary pair of a plaintext and a cipher text and then he finds the secret key using a quantum computer with computation in $O(\sqrt{2^n})$ steps. $O(\sqrt{2^n})$ steps is much better than the brute force attack which requires $O(2^n)$ steps.

[*]Communications Research Laboratory, 4-2-1 Nukui-Kitamachi Koganei Tokyo, 184-8795, Japan e-mail: aki@crl.go.jp
[†]Mitsubishi Electric Corp., 5-1-1 Ofuna Kamakura Kanagawa, 247-8501, Japan e-mail: ishizuka@isl.melco.co.jp

We also discuss the application of a quantum computer to the strength evaluation of a block cipher using Brassard, Høyer, Tapp's algorithm [10]. Their algorithm can be applied to compute non-uniformity of distribution between plaintexts, cipher texts and secret keys of a block cipher.

2 Grover's algorithm and Brassard, Høyer, Tapp's algorithm

2.1 Grover's algorithm

A. Definition of the problem

Suppose that there exist $N = 2^n$ states and each state is labeled by $0, 1, \dots, N-1$. We may assume that the states are represented by n bit binary strings. Let F be a Boolean function of the set of these 2^n states. Suppose that there exists a state w such that

$$F(x) = \begin{cases} 1 & \text{if } w = x \\ 0 & \text{otherwise.} \end{cases}$$

We suppose that F is effectively computable and the functional call to F is considered as an oracle call. The *searching problem* is that we can call oracle calls to F and find the state w. Clearly the exhaustive search algorithm needs $O(2^n)$ oracle queries on average to find the state w using a conventional computer.

B. Algorithm

1. Initialization: Applying Walsh-Hadamard transformation W $O(\log N)$ times to the initial state $|00\dots0\rangle$ bit by bit, we can obtain the following superposition
$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle.$$
2. Iteration: Repeat the following unitary operations M times.
   (a) For any states $|x\rangle$ in the present superposition, rotate the phase by π radians if $F(x) = 1$, leave the system unaltered, otherwise. See [15] for how to realize this operation.
   (b) Apply the diffusion transform $D(= D_{ij})$ defined by
   $$D_{ij} = \begin{cases} \frac{2}{N}, & \text{if } i \neq j \\ -1 + \frac{2}{N} & \text{if } i = j. \end{cases}$$
   Note that D is constructed by $D = WRW$ where, $R(= R_{ij})$ is the phase rotation matrix and $W(= W_{ij})$ is the Walsh-Hadamard transformation. R and W are defined by
   $$R_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j = 0 \\ -1, & i = j \neq 0 \end{cases} \quad \text{and} \quad W_{ij} = 2^{-n/2}(-1)^{\bar{i}\cdot\bar{j}},$$
   where $\bar{i} \cdot \bar{j}$ denotes the bitwise dot product.
3. Measurement: We measure the resulting superposition and get some state according to the probabilities determined by the amplitudes.

Proposition 2.1 ([13]) *After M iterations of of 2 (a), (b), the system is in the superposition*
$$\sin(2m+1)\theta|w\rangle + \cos(2m+1)\theta|b\rangle, \tag{2.1}$$
where θ is the angle satisfying $\sin\theta = \frac{1}{\sqrt{N}}$ and $|b\rangle$ is the superposition of bad states, that is, the states other than $|w\rangle$. □

See [9] for the proof. Proposition 2.1 implies the probability that we observe the state $|w\rangle$ in the process 3 above is $|\sin(2M+1)\theta|^2$. We can estimate the proper number of iterations of the process 2 to observe the desired state $|w\rangle$ by Proposition 2.1. It is easy to see that only $O(\sqrt{N})$ iterations are needed to observe the state $|w\rangle$ with a probability at least $\frac{1}{2}$.

In general, it is known that to find one of t desired states out of N states with a probability of at least $\frac{1}{2}$, we need iterate the process 2 (a), (b) $O(\sqrt{N/t})$ times (see [9]).

2.2 Brassard, Høyer, Tapp's algorithm

The quantum algorithm proposed by Brassard, Høyer, Tapp [10] counts the number of desired states in the whole space. Let $F : \{0,1,2,\dots,N-1\} \to \{0,1\}$ be a Boolean function. Suppose that $t = |F^{-1}(0)|$. The algorithm COUNT estimates $t$ calling $O(\sqrt{N})$ oracle queries to $F$. We should remark that it needs $O(N)$ oracle queries to $F$ on average to estimate $t$ using a conventional computer.

Shor's algorithm computes the period of a long sequence in polynomial time. Grover's algorithm finds a solution of the search problem in $O(\sqrt{N/t})$ steps. The algorithm *COUNT* is a combination of Shor's algorithm and Grover's algorithm. The rough idea of *COUNT* is the following. In Grover's algorithm, the amplitude for the desired states and the other states have a period depending upon the number $t$ of the desired states. We simulate Grover's algorithm and then compute the period by applying Shor's algorithm. Finding the period, we can compute the number $t$. The next lemma gives the error range of $t$. The number $P$ of trials of the processes 1 and 2 in Section 2.

Lemma 2.2 ([10]) *Let $F : \{0,1,2,\dots,N-1\} \to \{0,1\}$ be a Boolean function. COUNT calls oracle queries to $F$. Suppose $t = |F^{-1}(1)| \le \frac{N}{2}$. Let $\tilde{t}$ be the output of COUNT when we input the number $P$ ($P \ge 4$) of trials. Then we have*

$$|t-\tilde{t}| < \frac{2\pi}{P}\sqrt{tN} + \frac{\pi^2}{P^2}N \tag{2.2}$$

with probability $\frac{8}{\pi^2}D$

3 Cryptanalysis using Grover's algorithm

3.1 Simple attack scheme

We mount the known plaintext attack using Grover's algorithm. Let $E$ be a block cipher of key and block size $n$. Suppose that an attacker is given an arbitrary pair of a plaintext $P$ and the ciphertext $C$ encrypted by $E$ with a certain secret key k_j. Applying Grover's algorithm, he can find the secret key. For simplicity, we assume that there exists only one key k_j such that $C = E(P,k_j)$. As a matter of fact, there exist several keys satisfying $C = E(P,k_i)$ in general. In the case, the same analysis works using the method in [9], and so, we will not discuss it. Let us examine the details as follows.

1. Prepare a pair $(P,C)$ of a plaintext $P$ and a ciphertext $C$, that is, $C = E(P,k_j)$ for some secret key k_j. Define $F$ by
$$F(k_i) = \begin{cases} 1 & \text{if } E(P,k_i) = C \\ 0 & \text{if } E(P,k_i) \neq C \end{cases}$$
2. Initialize the system by making a superposition of all possible keys (k_i) with the same amplitude. Suppose the key length is 64 bit. There exist $N = 2^{64}$ states representing $N$ keys in the superposition. So we have the superposition
$$|K\rangle = \frac{1}{\sqrt{2^{64}}}\sum_{i=0}^{2^{64}-1}|k_i\rangle.$$
3. Iterate 2 (a) and (b) in Section 2.1. Note that k_j is the solution of the equation $F(k) = 1$. Possibly, $O(\sqrt{2^{32}})$ steps are needed.
4. Measuring the system, we observe the state $|k_j\rangle$ with a probability at least $\frac{1}{2}$, where k_j is the key satisfying $E(P,k_j) = C$.

Iterating 2 (a) and (b) $2^{32}$ times, we have

$$\begin{aligned} & G^{2^{32}}|K\rangle \\ \mapsto\ & \sin(2^{33}+1)\theta|k_g\rangle + \cos(2^{33}+1)\theta|k_b\rangle \\ \approx\ & \sin(\frac{2^{33}+1}{2^{32}})|k_g\rangle + \cos(\frac{2^{33}+1}{2^{32}})|k_b\rangle \\ \approx\ & 0.91|k_g\rangle - 0.42|k_b\rangle \end{aligned}$$

by (2.1). Note that since N is large enough, $\sin\theta$ can be approximated as θ, and hence,

$$\theta \simeq \sin\theta = \frac{1}{\sqrt{N}} = \frac{1}{\sqrt{2^{64}}}$$

As the probability is the square of the amplitude, the algorithm outputs the desired state with 83 percent($\approx 0.91^2$).

3.2 Exact number of steps for cryptanalysis

We now evaluate the number of iteration more carefully. Recall that the probability that we observe the desired state k_j is given by the square of the absolute value of the amplitude for w in (2.1). In our cryptanalysis, $w = k_j$. Remember that the amplitude for w is given by $\sin(2M+1)\theta$, where $\theta \approx \sin\theta$ and θ is defined by the equation $\sin\theta = \frac{1}{N}$. Hence, $\theta \approx \frac{1}{N}$. Solving the inequality

$$|\sin(2M+1)\theta| \geq \frac{1}{\sqrt{2}}, \quad 0 \leq \theta \leq \frac{\pi}{2},$$

we get

$$M \geq \pi 2^{\frac{n}{2}-3}.$$

In the case of cryptanalysis of a 64 bit block cipher, we need iterate the algorithm $M = \pi 2^{\frac{64}{2}-3}$ times. Then the number of iterations is slightly smaller than 2^{32}. In the case of cryptanalysis of a 128 bit block cipher, we need $M = \pi 2^{\frac{128}{2}-3}$ steps.

In [13], the amplitude k for desired state and the amplitude l for the other state are given by the following recurrence formula:

$$\begin{aligned} k_1 &= (\frac{2}{N} - 1)k + 2\frac{N-1}{N}l, \\ l_1 &= -\frac{2}{N}k + \frac{N-2}{N}l, \end{aligned} \tag{3.1}$$

where k_1 and l_1 are the amplitude after one iteration of 2 (a), (b). We compute the exact number of iterations using (3.1). These are close to the estimation based on (2.1) above.

n	N	M
4	16	1
8	256	5
12	4096	20
16	65536	84
20	1048576	337
24	16777216	1348
28	268435456	5394
36	68719476736	86308
38	274877906944	172616
*	343061989969	192840
40	1099511627776	345232
48	281474976710656	5523721

("*" is #4fe0168a51 by hexadecimal code)

3.3 Heuristic methods

Using a quantum computer, we break block ciphers more efficiently than the exhaustive search by conventional computers. The general clock speed of a current desktop personal computer is 800MHz. Using normal integer operations and some pipelines in computing, current commercial machines can process one clock per one operation. If 3 clocks are needed per operation, a present computer is still able to process about $\frac{8}{3} \approx 2.67$ hundred million times to 8 hundred million times per second. If the parallel processing scheme is employed, the processing power is improved. Therefore, $2^{32} = 4294967296$ steps are manageable. About 3 ($2^{28} = 268435456$) hundred million steps are needed for brute-force attack of DES. The computation can be carried out in one second or below.

In "DES challenge" sponsored by RSA Data Security Company, one group broke DES in only 22 hours using a brute force method distributing processes to numerous computers all over the world. They report that the correct key was found when 22 percent of the key space was searched.

Let us now discuss how we can reduce the computational cost using the detailed information on block ciphers. Consider searching a structured database. In a structured database, each data can be represented by a binary sequence. To obtain information from a structured database $O(\log N)$ steps are needed, where $N$ is the size of database. For example, we can find the desired data in 20 steps out of 1 million data in a structured database because $2^{20} = 1048576 > 1000000$.

In cryptanalysis of block ciphers, one employs the knowledge of non-uniformity of the distribution between plaintexts, secret keys and ciphertexts. Differential cryptanalysis and linear cryptanalysis [6, 14] are such cryptanalysis. Using non-uniformity of distributions, the size of key space can be reduced. If the size can be reduced to a tractable size, one can analyze by the brute-force method.

In quantum cryptanalysis, we may do the same. First, we reduce the size of the searching space by using the (possibly statistical) information on the searching space. Then we resort to quantum computers to do exhaustive search. Fig.1 schematically represents the idea.

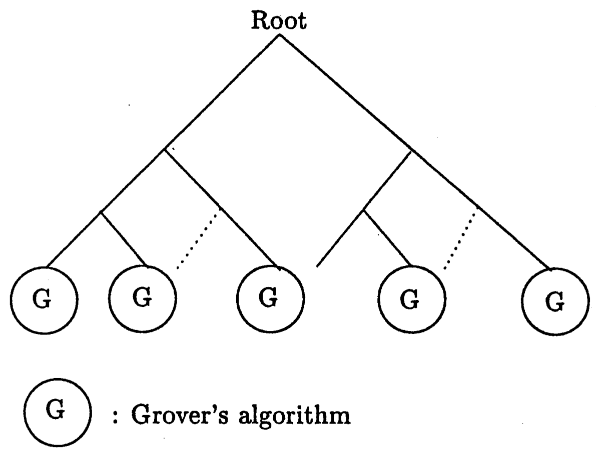

Fig.1 Applying Grover's algorithm partially

4 Strength evaluation using Brassard, Høyer, Tapp's algorithm

4.1 Strength evaluation of block ciphers

We discuss how to evaluate the statistical imbalance between plaintexts, secret keys and the corresponding ciphertexts using Brassard, Høyer, Tapp's algorithm. As we see in Section 2, *COUNT* algorithm estimates $|F^{-1}(0)|$ given the number of trials and a Boolean function $F : N \rightarrow \{0, 1\}$. Here, we assume $|F^{-1}(0)| < \frac{N}{2}$.

Suppose that $E$ is a block cipher. The key and block size may be arbitrary. Let us consider the probability ϕ_1 defined by

$$\phi_1 = \text{Prob}(P(i) \oplus K(j) = C(k)), \tag{4.1}$$

where $P(i)$ is the ith bit of the plaintext P, $K(j)$ is the jth bit of the secret key K, $C(k)$ is the kth bit of the corresponding ciphertext $C = E(P, K)$ and $\oplus$ is the exclusive or.

Let us also consider the probability ϕ_2 defined as follows:

$$\phi_2 = \text{Prob}(\Delta P(i) \oplus K(j) = \Delta C(k)), \tag{4.2}$$

where ΔP is the difference of two plaintexts, ΔC is the difference of two corresponding ciphertexts, that is, $\Delta P = P_1 \oplus P_2$, $\Delta C = C_1 \oplus C_2$, $C_1 = E(P_1, K)$, $C_2 = E(P_2, K)$. The probabilities (4.1) and (4.2) represent the non-uniformity of distribution between plaintexts, secret keys and ciphertexts. The reason that we are interested in the evaluating the probabilities (4.1) and (4.2) is that the modern cryptanalysis of block ciphers depend upon the statistical imbalance as we see in [6, 14].

The plausible scenario is that quantum computers will be established for billions of dollars and so only small number of machines will exist all over the world. It is impossible that desk-top size quantum computers will put on the commercial basis in the near future. A few quantum computers will be publicly owned by the governments. The users of the machines are limited to authorized research institutes and they can use the machines for only unclassified and peaceful purposes. The developers of symmetric ciphers may rental the machine to evaluate the strength of their ciphers not for other purposes.

4.2 How to compute ϕ_1

The probability ϕ_1 is not always substantially away from 1/2, however, if it is the case then one can collect sufficient amount of pairs of plaintexts and ciphertexts and mount a statistical cryptanalysis to find the secrete key. In such a situation, the block cipher E is not secure enough. Therefore, we require any block cipher not to have large statistical imbalances between plaintexts, ciphertexts and secrete keys.

To apply the *COUNT* algorithm, we define a Boolean function as follows. Suppose that E is a block cipher of key and block size 64. Denote the space of plaintexts by $\mathbf{P}$, the space of cichertexts by $\mathbf{C}$ and the space of secret keys by $\mathbf{K}$. Then both $\mathbf{P}$ and $\mathbf{K}$ consists of 2^{64} binary sequences of length 64. Let $\mathbf{D}$ be the direct product $\mathbf{P} \times \mathbf{K}$. Then $\mathbf{D}$ consists of 2^{128} binary sequences of length 128. Define the Boolean function F by

$$F(P, K) = \begin{cases} 0 & \text{if } P(i) \oplus K(j) = C(k) \\ 1 & \text{otherwise,} \end{cases} \tag{4.3}$$

where $1 \le i, j, k \le 64$. Suppose that $C = E(P, K)$. By the definition, the function F returns 0 for a pair (P, K) of a plaintext and a secret key that satisfies the equation

$$P(i) \oplus K(j) = C(k), \tag{4.4}$$

otherwise it returns 1. For the function F defined above, the probability ϕ_1 satisfies

$$\phi_1 = \frac{|F^{-1}(0)|}{|N|} = \frac{2^{128} - t}{2^{128}}.$$

We consider finding the integer $|F^{-1}(0)|$ using *COUNT* algorithm.

Let $\tilde{t}$ be the output of *COUNT* algorithm given F and P. Then the inequality (2.2) holds with the probability $8/\pi^2$. Therefore, repeating *COUNT* algorithm, we can obtain a good estimation $\tilde{t}$. The error, the right hand side of (2.2), depends upon only P. For example, let $P = 2^{32}$. Then we have

$$|t - \tilde{t}| \le \frac{2\pi}{2^{32}}\sqrt{2^{128}t} + \frac{\pi^2}{(2^{32})^2}2^{128} \le 2^{96}\sqrt{2}\pi + 2^{64}\pi^2$$

with the probability $8/\pi^2$, because $t \le \frac{2^{128}}{2} = 2^{127}$. Hence we have

$$\phi_1 = \frac{|F^{-1}(0)|}{|N|} = \frac{|F^{-1}(0)|}{2^{128}} = \frac{2^{128} - t}{2^{128}} = 1 - \frac{t}{2^{128}} \approx 1 - \frac{\tilde{t}}{2^{128}} = \tilde{\phi}_1$$

and the error is estimated as

$$|\phi_1 - \tilde{\phi}_1| = \frac{|t - \tilde{t}|}{2^{128}} \le \frac{2^{96}\sqrt{2}\pi + 2^{64}\pi^2}{2^{128}}.$$

Since we have

$$\frac{2^{96}\sqrt{2}\pi + 2^{64}\pi^2}{2^{128}} < \frac{\pi}{2^{30}},$$

the error $|\phi_1 - \tilde{\phi}_1|$ is sufficiently small. Hence, we can estimate ϕ_1 as the relative probability of the pair of plaintexts and secret keys (P, K) satisfying the equation (4.4) with respect to the whole space. In this case, about 2^{32} iterations of *COUNT* algorithm is required.

Similarly, if we carry out the experiment with $P = 2^{16}$ iterations of *COUNT* algorithm, then we can estimate ϕ_1 with the probability $\frac{8}{\pi^2}$ within error $\frac{\sqrt{2}\pi}{2^{16}} + \frac{\pi^2}{2^{32}} \approx 0.0001$.

4.3 How to compute ϕ_2

We next estimate the probability ϕ_2. Let **P**, **K** and **C** be the space of plaintexts, secrete keys and ciphertexts, respectively. Let $\mathbf{D} = \mathbf{P} \times \mathbf{P} \times \mathbf{K}$. Then **D** consists of 2^{192} binary sequences of length 192. Define $F : \mathbf{D} \to \{0, 1\}$ as follows.

$$F(P_1, P_2, K) = \begin{cases} 0 & \text{if } \Delta P(i) \oplus K(j) = \Delta C(k) \\ 1 & \text{otherwise,} \end{cases} \tag{4.5}$$

where $1 \le i, j, k \le 64$, $\Delta P = P_1 \oplus P_2$, $\Delta C = C_1 \oplus C_2$, $C_1 = E(P_1, K)$ and $C_2 = E(P_2, K)$.

By the definition, the function F returns 1 for a triple (P_1, P_2, K) satisfying

$$\Delta P(i) \oplus K(j) = \Delta C(k), \tag{4.6}$$

otherwise it returns 0. Let $P = 2^{32}$ and $t = |F^{-1}(1)|$. Let $\tilde{t}$ be the output of *COUNT* algorithm. A similar argument to the one above shows that

$$|t - \tilde{t}| \le \frac{2\pi}{2^{32}}\sqrt{2^{192}t} + \frac{\pi^2}{(2^{32})^2}2^{192} \le 2^{160}\sqrt{2}\pi + 2^{128}\pi^2.$$

Hence, we have

$$\phi_2 = \frac{|F^{-1}(0)|}{|N|} = \frac{|F^{-1}(0)|}{2^{192}} = \frac{2^{192} - t}{2^{192}} = 1 - \frac{t}{2^{192}} \approx 1 - \frac{\tilde{t}}{2^{192}} = \tilde{\phi}_2$$

within the error

$$|\phi_2 - \tilde{\phi}_2| = \frac{|t - \tilde{t}|}{2^{192}} \le \frac{2^{160}\sqrt{2}\pi + 2^{128}\pi^2}{2^{192}}.$$

Since

$$\frac{2^{160}\sqrt{2}\pi + 2^{128}\pi^2}{2^{192}} < \frac{\pi}{2^{30}},$$

the error $|\phi_2 - \tilde{\phi}_2|$ is sufficiently small.

If we carry out the experiment with $P = 2^{16}$, then we can estimate ϕ_2 with the probability $\frac{8}{\pi^2}$ within the error $\frac{\sqrt{2}\pi}{2^{16}} + \frac{\pi^2}{2^{32}} \approx 0.0001$.

4.4 Number of trials

The error is $\frac{2\pi}{P}\sqrt{tN} + \frac{\pi^2}{P^2}N$ when *COUNT* algorithm is applied to find $|F(1)^{-1}|$ with the probability $\frac{8}{\pi^2}$. It depends upon the number P of trials and the cardinality $N = 2^n$ of the domain of F. We calculated the error in cryptanalysing block ciphers of block size 64 bit as above. It is easy to see that we can estimate the probability ϕ_1 for 128 bit block cipher within the error $\frac{\pi}{2^{30}}$ for the probability $\frac{8}{\pi^2}$ by iterating $P = 2^{32}$ times. This indicates that the increase of the size of block does not affect the number of trials as far as we fix the upper bound of the error. The following can be easily proved.

Theorem 4.1 *For any block cipher E, we have*

$$|\phi_1 - \tilde{\phi}_1| \le \frac{\sqrt{2}\pi}{P} + \frac{\pi^2}{P^2}.$$

4.5 Remarks

We discussed how to estimate statistical imbalances for block ciphers using quantum computers. Although we considered probabilities given in (4.4), (4.6), statistical imbalances of block ciphers appear in various forms. The algorithm given above does not clarify some other type of statistical imbalances. However, combining traditional cryptanalysis technologies and quantum computers, we can estimate statistical imbalance and evaluate the strength of block ciphers. It is desired that we can invent an algorithm that automatically finds non-uniformity of distributions on plaintexts, secret keys and ciphertexts. A possible approach along this line is to run Brassard, Høyer, Tapp's algorithm in Grover's algorithm as subroutines. It is not clear whether it is possible or not without substantial speed-downs.

5 Summary

We discussed how to apply quantum computers for cryptanalysis and strength evaluation of block ciphers. The realization of quantum computers depends upon the technologies for the control of decoherence of quantum states. We need complete understanding of the entanglement. It seems premature to discuss the realization of quantum computers. The role of quantum physics in information and computer science gets bigger and bigger from now on. Quantum physical concepts like interference, the uncertainty principle, the measurement are essential in quantum information processing. In information security, quantum physics will play a significant role in the future as we see quantum cryptography gets huge attention.

In addition, quantum cryptography [4], quantum protocols and quantum teleportation are implemented using quantum phenomena such as the EPR effects, the uncertainty principle and the entanglement.

Another non-classical computation device is implemented by Adleman [1] using DNA, that is, molecular computing. He succeeded in finding a solution for the traveling salesman problem of small size. In [7], cryptanalysis of DES using DNA computing is considered.

All these trials indicate that physical phenomena can be employed to create new technologies that are not realized by traditional methods. The theory of computation is based on not only mathematical method but physics and other natural science. In the theory of quantum computing, the realization of quantum computers is crucial, however, developments of rigorous theory on quantum algorithms from the point of view of computational complexity is also inevitable for good understanding and more future applications.

References

[1] L.Adleman, "Molecular computation of solutions to combinatorial problems", Science **266** November (1994), 1021-1024.

[2] C.H.Bennett, "Logical reversibility of computation", IBM J. Research and Development, **6** (1973), 525-532.

[3] C.H.Bennett, E.Bernstein, G.Brassard and U.Vazirani, "Strengths and weaknesses of quantum computing", SIAM J. Comp. (5) **26** (1997), 1510-1523.

[4] C.H.Bennett and G.Brassard, "Quantum cryptography: Public-key distribution and coin tossing", Proc. Int. Conf. on computers, Systems and Signal Processing, Bangalore, India (1984), 175-179.

[5] E.Bernstein and U.Vazirani, "Quantum complexity theory", SIAM J. Comp. (5) **26** (1997), 1411-1473.

[6] E.Biham and A.Shamir, "Differential cryptanalysis of DES-like cryptosystems", Advances in Cryptology (CRYPTO'90), LNCS **537** (1991), 2-21.

[7] D.Boneh, C.Dunworth and R.J.Lipton "Breaking DES using a molecular biology" in DNA Based Computers, Proc. of DIMACS workshop, American Mathematics Society, (1995), 37-65.

[8] D.Boneh and R.J.Lipton, "Quantum Cryptanalysis of Hidden Linear Functions", Advances in Cryptology (CRYPTO'95), LNCS **921** (1995), 424-437.

[9] M.Boyer,G.Brassard,P.Høyer and A.Tapp, "Tight bounds on quantum searching", PhysComp'96 (1996), 36-43.

[10] G.Brassard,P.Høyer and A.Tapp, "Quantum Counting", Int. Coll. Automata, Language and Programming (ICALP'98), LNCS **1443** (1998), 820-831.

[11] R.P.Feynmann, "Simulating physics with computers", International J. Theoretical Phys., **21** (1982), 467-488.

[12] R.P.Feynmann, "Quantum mechanical computers", Optics News, **11** (1985), 11-20.

[13] L.K.Grover, "A fast quantum mechanical algorithm for database search", Proc. 28th ACM Symp. on Theory of Computation (STOC'96) (1996), 212-219.

[14] M.Matsui, "The first experimental cryptanalysis of the Data Encryption Standard", Advances in Cryptology (CRYPTO'94), LNCS **839** (1994), 1-11.

[15] J.Preskill, "Quantum Information and Computation", Lecture Notes for Physics **229** California Institute of Technology (1998).

[16] P.W.Shor, "Polynomial time algorithm for prime factorization and discrete logarithms on a quantum computer", SIAM J. Comp. (5) **26** (1997), 1484-1509.

[17] A.Yao, "Quantum circuit complexity", Proc. 34th Annual Symp. on Foundation of Computer Science (FOCS'93) (1993), 352-361.